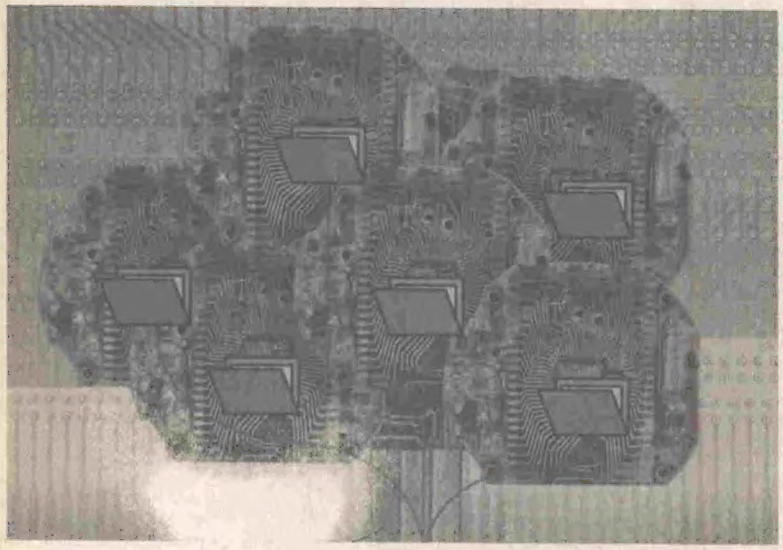
ยุคลอาจ ชาญพานิชกิจการ

การใช้ประโยชน์จากไฟล์ .INI ด้วย Visual BASIC



เรื่องของการใช้ประโยชน์จากไฟล์ .INI ว่าจะใช้ประโยชน์อะไรได้บ้าง ด้วยการสร้างโปรแกรมโดยใช้ Visual BASIC

สวัสดีครับ สำหรับบทความตอนนี้จะกล่าวถึงเรื่องของ INI ไฟล์ ซึ่งจะพูดถึงเรื่องของการใช้ประโยชน์จากไฟล์นี้ว่าเราจะใช้ประโยชน์อะไรได้บ้าง ตัวอย่างของบทความตอนนี้ผมได้สร้างโปรแกรมขึ้นมาโดยใช้ Visual BASIC เป็นตัวทำงาน โดยจะเรียกเอาค่าต่าง ๆ ที่เราเก็บเอาไว้มาใช้งาน และสามารถกำหนดค่าต่าง ๆ เก็บเอาไว้ได้ การที่เราจะใช้ค่าต่าง ๆ จาก INI ไฟล์นั้น ถ้าเราจะเขียนโปรแกรมขึ้นมาจัดการในส่วนนี้เองก็ย่อมทำได้ แต่ว่าเราสามารถที่จะใช้ API ของวินโดว์ได้เลย โดยจะใช้ฟังก์ชันที่มีชื่อว่า GetProfileString เพื่อที่จะดึงค่าต่าง ๆ ใน INI ไฟล์มาใช้งาน ตัวอย่างโปรแกรมนี้เมื่อคุณมีการแก้ไขความกว้างความยาวของฟอร์มสีของฟอร์ม หรือรูปแบบการแสดงของฟอร์มแล้ว เมื่อมีการบันทึกค่านี้เก็บเอาไว้และมีการ Load INI ขึ้นมาใหม่ โปรแกรมจะทำการปรับฟอร์มไปตามรูปแบบที่เราได้ทำการกำหนดเอาไว้ ไม่ว่าจะมีการเรียกโปรแกรมนี้ขึ้นมาใหม่ก็จะได้ค่าตามที่เรากำหนดเอาไว้ สิ่งนี้เป็นตัวอย่างจากการใช้ประโยชน์จาก INI ไฟล์ ตัวโปรแกรมจะเก็บค่าตามส่วนต่าง ๆ ดังนี้

- เก็บขนาดของฟอร์ม
- เก็บสีของฟอร์ม
- WindowsState
- Current Path

การใช้งานส่วนนี้สามารถใส่ความสูงและความกว้างได้ตามที่ต้องการ (ดูรูปที่ 1)

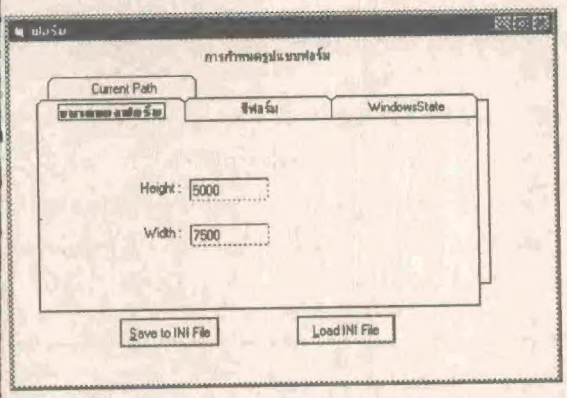
การกำหนดสีของฟอร์มเราสามารถกำหนดทั้งหน้าและสีพื้นหลัง (ดูรูปที่ 2)

การกำหนดรูปแบบการแสดงของฟอร์ม เราสามารถที่จะกำหนดได้ว่าจะให้แสดงแบบธรรมดา แบบเล็กสุด แบบเต็มจอ (ดูรูปที่ 3)

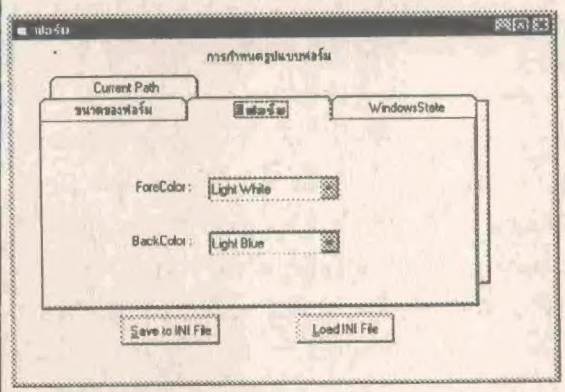
และส่วนสุดท้ายที่สำคัญอีกจุดก็คือการกำหนด PATH ของโปรแกรมซึ่งเราสามารถที่จะเก็บค่านี้ไว้ได้ตลอด อย่างในกรณีที่ต้องมีการใช้ดาต้าเบสต่าง ๆ เราก็สามารถที่จะระบุได้ (ดูรูปที่ 4)

ลักษณะการเก็บข้อมูลของ INI ไฟล์

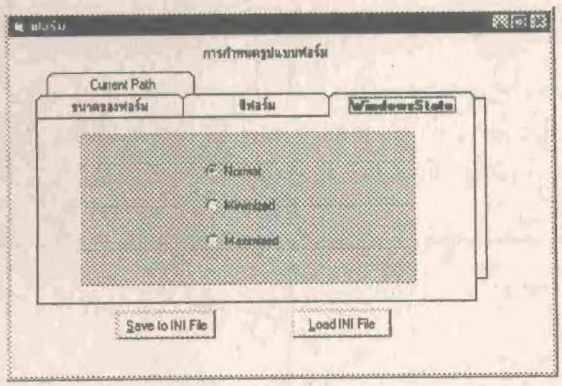
จากข้อกำหนดที่เราจะให้เก็บค่าต่าง ๆ เหล่านี้ เราจะมาแบ่งรูปแบบการเก็บข้อมูลดังนี้ อย่างแรกต้องมี SECTION ซึ่งเรากำหนดให้เป็น [Apps] ภายใต้ SECTION จะเก็บค่าต่าง ๆ เอาไว้ดังตารางที่ 1



รูปที่ 1 แสดงการเก็บข้อมูลความกว้างและความสูงของฟอร์ม



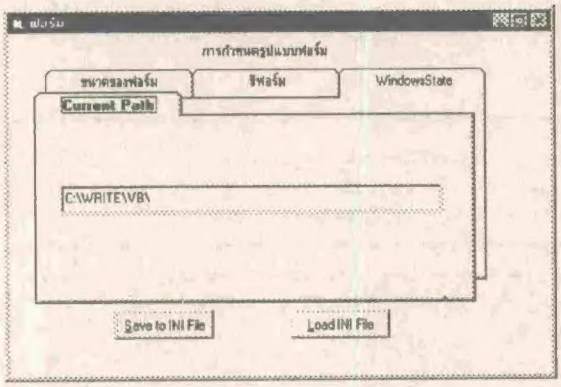
รูปที่ 2 แสดงการเก็บสีของฟอร์ม



รูปที่ 3 WindowsState

การเก็บข้อมูลในไฟล์ TEST.INI จะเก็บในลักษณะนี้

```
File : TEST.INI
[Apps]
BackColor=9
ForeColor=14
Width=7500
Height=5000
WindowState=0
Path=C:\WRITE\VB\EVS
```



รูปที่ 4 แสดง PATH ของโปรแกรม

ชื่อ	ความหมาย
BackColor	สีพื้นฟอร์ม
ForeColor	สีหน้าฟอร์ม
Width	ความกว้างของฟอร์ม
Height	ความสูงของฟอร์ม
WindowState	รูปแบบการแสดงของฟอร์ม
Path	ที่เก็บข้อมูล

ตารางที่ 1

การสร้างโปรแกรม

อันดับแรกเลยนั้นเราจะต้องทำการประกาศชนิดของฟังก์ชัน GetPrivateProfileString (API Function) ก่อนใน Module รูปแบบการประกาศมีดังนี้

```
Declare Function GetPrivateProfileString Lib "kernel32" _
 Alias "GetPrivateProfileStringA" _
 (ByVal lpApplicationName As String, _
 ByVal lpKeyName As Any, _
 ByVal lpDefault As String, _
 ByVal lpReturnedString As String, _
 ByVal nSize As Long, _
 ByVal lpFileName As String) As Long
```

สร้างฟอร์มขึ้นมาหนึ่งฟอร์มมีออบเจ็กต์ SSTAB โดยมีจำนวน TAB ทั้งหมด 4 TAB ในแต่ละ TAB จะเก็บข้อมูลแตกต่างกันออกไป เมื่อเราต้องการแก้ไขข้อมูลในส่วนใด ๆ ของฟอร์มเราสามารถที่จะกำหนดค่าเอาไว้ได้เลย เมื่อมีการเรียกฟอร์มนี้ขึ้นมาเมื่อไรก็จะได้ค่าที่เราได้กำหนดเอาไว้ให้ ฉะนั้นในส่วนของอีเวนต์ฟอร์ม Load เราจะต้องกำหนดให้ทำการ Load ข้อมูลพรอพเพอร์ตี้ต่าง ๆ ของฟอร์มขึ้นมาเพื่อที่จะกำหนดรูปแบบของฟอร์มตามที่เราได้กำหนดเอาไว้ ดูตัวอย่างต่อไปนี้ อันดับแรกกำหนดตัวแปรขึ้นมารับข้อมูลค่าต่าง ๆ ก่อน เราจะใช้

ฟังก์ชัน GetProfile ดึงค่าที่เก็บไว้ใน INI ไฟล์มาใช้ วิธีการใช้ GetProfile มีรูปแบบการใช้งานดังนี้

```
Syntax :
      GetProfile(cSECTION,cKEY)
      - cSECTION      ส่วน
      - cKEY          ชื่อของค่าที่ต้องการดึงข้อมูล
```

```
[Apps]      <-- cSECTION
BackColor=9 <-- cKEY
```

อีเวนต์ Form_Load

```
Private Sub Form_Load()
Dim xfcolor, xbcolor, nwidth, nheight, nWindow, ccpath
 xbcolor = GetProfile("Apps", "BackColor")
 xfcolor = GetProfile("Apps", "ForeColor")
 nwidth = GetProfile("Apps", "Width")
 nheight = GetProfile("Apps", "Height")
 nWindow = GetProfile("Apps", "WindowState")
 ccpath = GetProfile("Apps", "Path")
```

หลังจากที่ดึงค่าจาก INI ไฟล์แล้วเราจะเพิ่มรายการสีเพื่อให้ผู้ใช้สามารถกำหนดสีได้ตามต้องการ ทั้งสี BackGround กับ ForeGround โดยออบเจ็กต์ทั้งสองมีชื่อว่า fColor กับ bColor

```
 fcolor.AddItem "Blak"
 fcolor.AddItem "Blue"
 fcolor.AddItem "Green"
 fcolor.AddItem "Cyan"
 fcolor.AddItem "Red"
 fcolor.AddItem "Magenta"
 fcolor.AddItem "Yellow"
 fcolor.AddItem "White"
 fcolor.AddItem "Gray"
 fcolor.AddItem "Light Blue"
 fcolor.AddItem "Light Green"
 fcolor.AddItem "Light Cyan"
 fcolor.AddItem "Light Red"
 fcolor.AddItem "Light Magenta"
 fcolor.AddItem "Light Yellow"
 fcolor.AddItem "Light White"
 bcolor.AddItem "Blak"
 bcolor.AddItem "Blue"
 bcolor.AddItem "Green"
 bcolor.AddItem "Cyan"
 bcolor.AddItem "Red"
 bcolor.AddItem "Magenta"
 bcolor.AddItem "Yellow"
 bcolor.AddItem "White"
 bcolor.AddItem "Gray"
 bcolor.AddItem "Light Blue"
 bcolor.AddItem "Light Green"
 bcolor.AddItem "Light Cyan"
 bcolor.AddItem "Light Red"
 bcolor.AddItem "Light Magenta"
 bcolor.AddItem "Light Yellow"
 bcolor.AddItem "Light White"
 fcolor.ListIndex = xfcolor
 bcolor.ListIndex = xbcolor
```

```
 txtWidth.Text = nwidth
 txtHeight.Text = nheight
```

คำสั่ง Select Case นี้ใช้ในการตรวจสอบว่า Window ที่ต้องการให้แสดงอยู่ในรูปแบบใด โดยตรวจสอบจากค่าของ nWindow ค่าของ nWindow จะอยู่ระหว่าง 0-2 ถ้าค่าของ nWindow เป็น 0 จะมีความหมายเท่ากับ Normal ซึ่งจะต้องให้ค่า OPTION1 เป็น True ถ้าค่า nWindow เป็น 1 หมายความว่า Minimized ค่าของ OPTION2 จะเท่ากับ True แทน และสุดท้ายก็คือ OPTION3 จะมีค่าเท่ากับ Maximized ตามลำดับ

```
Select Case Val(nWindow)
      Case 0
            Option1.Value = True
      Case 1
            Option2.Value = True
      Case 2
            Option3.Value = True
End Select
```

demo เป็นชื่อของฟอร์ม ฉะนั้นเมื่อมีการอ่านค่าสีขึ้นมาแล้วเราจะต้องกำหนดสีให้กับฟอร์มด้วย เราจะใช้ฟังก์ชัน QBColor ในการกำหนดสีให้กับฟอร์ม สำหรับฟังก์ชันนี้จะต้องใช้ตัวเลขในการกำหนดสี ฉะนั้นในการทำงานเราจะต้องบันทึกข้อมูลสีเป็นตัวเลข ค่าสีนั้นจะมีค่าเป็นตัวเลขเรียงลำดับกันมาตามตำแหน่งของ ListIndex ใน Combo Box ตัวอย่างเช่น 0 เท่ากับสีดำ 1 มีค่าเท่ากับสีน้ำเงิน

```
 demo.ForeColor = QBColor(Val(xfcolor))
 demo.BackColor = QBColor(Val(xbcolor))
 demo.Height = Val(nheight)
 demo.Width = Val(nwidth)
 demo.WindowState = Val(nWindow)
```

สำหรับ TAB ออบเจ็กต์เราจำเป็นที่จะต้องกำหนดสีให้เหมือนกับฟอร์มด้วยเพื่อความสวยงาม และออบเจ็กต์ Label ที่แสดงก็จำเป็นจะต้องถูกกำหนดในลักษณะเดียวกันด้วย

```
 SSTab1.BackColor = QBColor(Val(xbcolor))
 SSTab1.ForeColor = QBColor(Val(xfcolor))
 Label5.BackColor = QBColor(Val(xbcolor))
 Label5.ForeColor = QBColor(Val(xfcolor))
 cPath.Text = ccpath

End Sub
```

ปุ่มบันทึก (Save)

เมื่อผู้ใช้สั่งบันทึกค่าลงใน INI ไฟล์เราจะต้องใช้ฟังก์ชันเขียนข้อมูลลงไปใน INI ไฟล์ เนื่องจากว่าเรามีการเก็บข้อมูลทุก ๆ ตัว ในการเปิด INI ไฟล์เราจึงใช้วิธีในการสร้าง INI ไฟล์ใหม่ทุกครั้ง แต่สำหรับการใช้ INI ไฟล์เก็บข้อมูลมาก ๆ การให้โปรแกรมสร้าง INI ไฟล์ใหม่

ทุกครั้งจะไม่สะดวกเพราะจะต้องมีตัวแปรมารองรับกับค่าต่าง ๆ นี้ วิธีที่ดีสำหรับการที่เราต้องมีการเก็บค่าที่มีจำนวนมากนี้ควรจะใช้ API ฟังก์ชัน ที่มีชื่อว่า WritePrivateProfileString จะดีกว่า

```
Private Sub But_Save_Click()
      Open "C:\WIN95\TEST.INI" For Output As #1
      Print #1, "[Apps]"
      Print #1, "BackColor=" & bcolor.ListIndex
      Print #1, "ForeColor=" & fcolor.ListIndex
      Print #1, "Width=" & txtWidth.Text
      Print #1, "Height=" & txtHeight.Text
      If Option1.Value Then
            Print #1, "WindowState=" & "0"
      End If
      If Option2.Value Then
            Print #1, "WindowState=" & "1"
      End If
      If Option3.Value Then
            Print #1, "WindowState=" & "2"
      End If
      Print #1, "Path=" & CurDir()
      Close #1
End Sub
```

สำหรับอีเวนต์ Load นี้ จะมีวิธีการเหมือนกับฟอร์ม Load ซึ่งจะให้อีเวนต์ Form_Load มาเรียกอีเวนต์ But_Load_Click แทนก็ได้

```
Private Sub But_Load_Click()
      Dim xfcolor, xbcolor, nwidth, nheight, nWindow
            nWindow = 0
            Unload demo
            xbcolor = GetProfile("Apps", "BackColor")
             xfcolor = GetProfile("Apps", "ForeColor")
            nwidth = GetProfile("Apps", "Width")
            nheight = GetProfile("Apps", "Height")
            nWindow = GetProfile("Apps", "WindowState")
            ccpath = GetProfile("Apps", "Path")
            demo.ForeColor = QBColor(Val(xfcolor))
            demo.BackColor = QBColor(Val(xbcolor))
            demo.Height = Val(nheight)
            demo.Width = Val(nwidth)
            demo.WindowState = Val(nWindow)
            cPath.Text = ccpath
            demo.Show
End Sub
```

ก่อนที่คุณจะใช้ฟังก์ชัน API ต่าง ๆ ได้คุณจะต้องประกาศก่อนว่ามาจาก Lib ใดและมีอาร์กิวเมนต์อะไรบ้าง

```
Module
Option Explicit
Const ENUM_BUF_SIZE = 4096
```

สำหรับอาร์กิวเมนต์ในการประกาศนั้นให้ดูจาก API Viewer (เครื่องมือของ Visual Basic) ว่ามีอะไรบ้าง

```
Declare Function GetPrivateProfileString Lib "kernel32" _
 Alias "GetPrivateProfileStringA" _
 (ByVal lpApplicationName As String, _
 ByVal lpKeyName As Any, _
 ByVal lpDefault As String, _
 ByVal lpReturnedString As String, _
 ByVal nSize As Long, _
 ByVal lpFileName As String) As Long
```

สำหรับการใช้งานของเราการส่ง Parameter ในลักษณะนี้คงจะไม่สะดวกเพราะว่ามีจำนวนมาก และค่าต่าง ๆ เราสามารถที่จะหาได้จากการนำค่าเพียงสองค่าที่ส่งมานี้มาหาได้ เราจึงต้องสร้างฟังก์ชันขึ้นมาเป็นตัวรับอีกที โดยเพราะการกระชับชื่อของฟังก์ชัน

```
Function GetProfile(sSection As String, sKey As String) As
String
 Dim retLen As Integer
 Dim buf As String * ENUM_BUF_SIZE
 retLen = GetPrivateProfileString(sSection, sKey, "", buf,
ENUM_BUF_SIZE, "C:\WIN95\TEST.INI")
 If retLen = 0 Then
 GetProfile = ""
 Else
 GetProfile = Left$(buf, retLen)
 End If
End Function
```

ตัวแปรสำหรับรับค่าจาก INI ไฟล์

ตำแหน่ง INI ไฟล์